Facultad de Informática de Madrid

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL - 1ª EVALUACIÓN (21 de enero de 2016)

Apellidos: SOLUCIÓN Nombre:

Ejercicio 1:

Dada una gramática de tipo 2:

a) Definir derivación por la izquierda de una palabra.] TEORIA
Definir gramática ambigua.

b) Comprobar si es ambigua o no lo es la gramática:

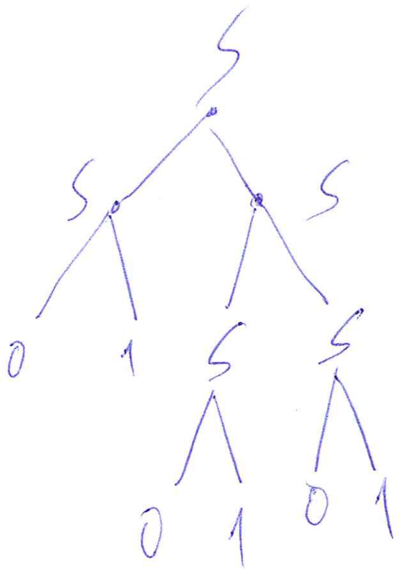
G = { Σ_T = { 0 , 1 }, Σ_N = { S }, S, $\mathcal{P}$} con las siguientes producciones:

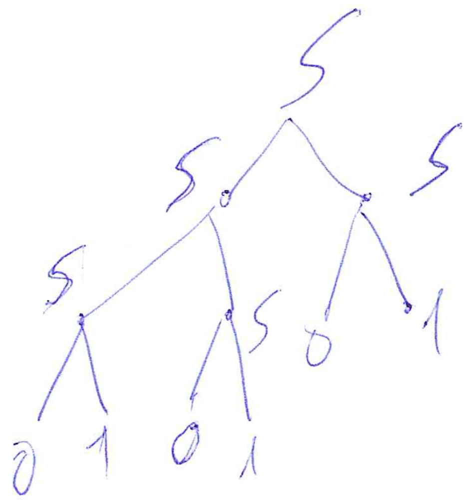
$\mathcal{P} \equiv \big|$ S : : = SS | 01

25 minutos

---

b/ Una gramática es ambigua si existe una palabra ambigua. G es ambigua porque x es palabra ambigua. Al tener dos árboles de derivación diferentes



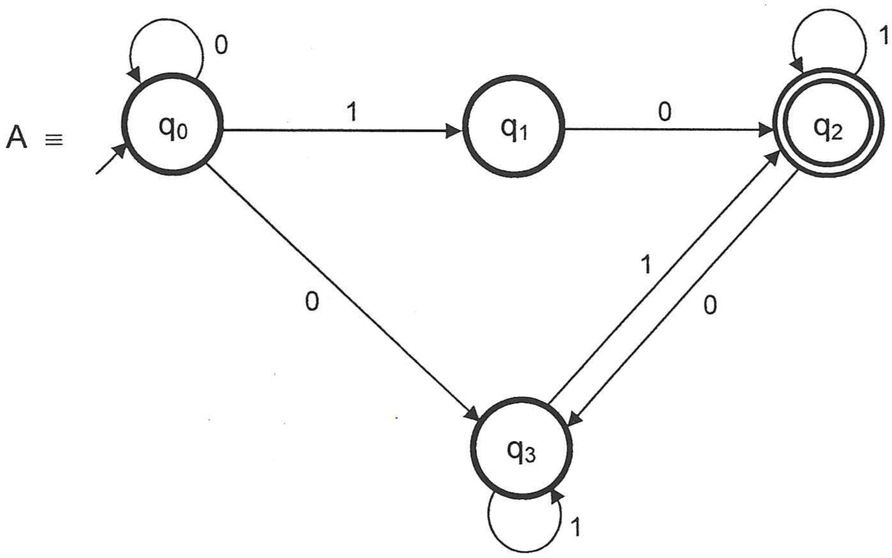


Facultad de Informática de Madrid
LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD
FINAL - 1ª EVALUACIÓN (21 de enero de 2016)

Apellidos: SOLUCIÓN Nombre:

Ejercicio 2:

Dado el autómata finito A, descrito mediante el siguiente diagrama de estados, obtener mediante ecuaciones características el lenguaje reconocido por dicho autómata.



25 minutos

$$\begin{cases} X_0 = 0X_0 + 1X_1 + 0X_3 \\ X_1 = 0X_2 \\ X_2 = 1X_2 + 0X_3 + \lambda \\ X_3 = 1X_3 + 1X_2 \end{cases}$$

$$X_3 = 1X_3 + 1X_2 = 1^*1X_2$$

$$X_2 = 1X_2 + 0X_3 + \lambda = 1X_2 + 01^*1X_2 + \lambda$$

$$X_2 = (1 + 01^*1)X_2 + \lambda; \quad \underline{X_2 = (1+01^*1)^*}$$

$$X_1 = 0X_2; \quad \underline{X_1 = 0(1+01^*1)^*}$$

$$X_0 = 0X_0 + 1X_1 + 0X_3 = 0X_0 + 10(1+01^*1)^* + 0X_3 =$$

$$= 0X_0 + 10(1+01^*1)^* + 01^*1X_2 = 0X_0 + 10(1+01^*1)^* +$$

$$+ 01^*1(1+01^*1)^*$$

$$L(A) = X_0 = 0^*\left(10(1+01^*1)^* + 01^*1(1+01^*1)^*\right)$$

Apellidos: SOLUCIÓN Nombre:

Ejercicio 1:

Dada la Gramática G = { Σ_T, Σ_N, S, $\mathcal{P}$ } donde Σ_T = { 0, 1 }, Σ_N = { S, A, B, C }, S = axioma y $\mathcal{P}$ las producciones:

S : : = 0AC | 0BCC | 0C | 0CC | λ
A : : = 0AC | 0C
B : : = 0BCC | 0CC
C : : = 1

a) Construir un Autómata a Pila (AP) que reconozca el lenguaje generado por dicha gramática. Utilizar, únicamente, uno de los 2 métodos conocidos (7 puntos).

b) Comprobar la aceptación de las palabras 0011 y 001111 en el AP (2 puntos) y la generación de las mismas en la Gramática (1 punto).

25 minutos

METODO 1

Construir un AP que acepte el lenguaje generado por la gramática: G = { Σ_T, Σ_N, $\mathcal{P}$, S }.
La gramática ha de estar en FNG.

AP = { Σ_T , Σ_N , { q } , S , q , f , $\varnothing$ }

Σ_T = alfabeto de entrada (Σ)
Σ_N = alfabeto de pila (Γ)
{ q } = Q conjunto de estados de AP
S = Símbolo de inicio de pila

q = estado inicial de AP
F = Función de transición (movimientos)
F = $\varnothing$

ALGORITMO (para obtener los movimientos del AP):

1. Si A : : = aZ , $\forall$ a $\in \Sigma_T$, A $\in \Sigma_N$, Z $\in \Sigma_N^*$ entonces, (q , Z) $\in f$ (q , a , A)
2. Si S : : = λ entonces, (q , λ) $\in f$ (q , λ , S)

SOLUCION MÉTODO 1:

Se va a construir un AP que acepte el lenguaje generado por la gramática G = { { a, b }, { S, A, B }, $\mathcal{P}$, S }
La gramática ya está en FNG.

AP = { Σ_T , Σ_N , { q } , S , q , f , $\varnothing$ } , AP = { { 0, 1 }, { S, A, B, C } , { q } , S , q , f , $\varnothing$ }

Aplicamos el ALGORITMO para obtener los movimientos del AP:

f (q , 0 , S) = { (q , AC) , (q , BCC) , (q , C) , (q , CC) }
f (q , λ , S) = { (q , λ) }
f (q , 0 , A) = { (q , AC) , (q , C) }
f (q , 0 , B) = { (q , BCC) , (q , CC) }
f (q , 1 , C) = { (q , λ) }

RECONOCIMIENTO AP (0011): (q , 0011 , S) $\vdash$ (q , 011 , AC) $\vdash$ (q , 11 , CC) $\vdash$ (q , 1 , C) $\vdash$ (q , λ , λ)

RECONOCIMIENTO AP (001111): (q , 001111 , S) $\vdash$ (q , 01111 , BCC) $\vdash$ (q , 1111 , CCCC) $\vdash$ (q , 111 , CCC) $\vdash$ (q , 11 , CC) $\vdash$ (q , 1 , C) $\vdash$ (q , λ , λ)

GENERACION G (0011): S $\rightarrow$ 0AC $\rightarrow$ 00CC $\rightarrow$ 001C $\rightarrow$ 0011
GENERACION G (001111): S $\rightarrow$ 0BCC $\rightarrow$ 00CCCC $\rightarrow$ 001CCC $\rightarrow$ 0011CC $\rightarrow$ 00111C $\rightarrow$ 001111

Facultad de Informática de Madrid

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL – 2ª EVALUACIÓN (21 de enero de 2016)

Apellidos: Nombre:

METODO 2

Construir un AP que acepte (reconozca) el lenguaje generado por la gramática: G = { Σ_T, Σ_N, $\mathcal{P}$, S }. La gramática no ha de estar necesariamente en FNG.

AP = { Σ_T , { $\Sigma_N \cup \Sigma_T$ } , { q } , S , q , f , $\varnothing$ }

Σ_T = Alfabeto de entrada (Σ)	q = estado inicial del AP
{ $\Sigma_N \cup \Sigma_T$ } = Alfabeto de pila (Γ)	F = Función de transición (movimientos)
{ q } = Q (Conjunto de estados del AP)	F = $\varnothing$ (Conjunto de estados finales)
S = Símbolo de inicio de pila	

ALGORITMO (para obtener los movimientos del AP):

1. X $\in$ { $\Sigma_N \cup \Sigma_T$ }, A $\in \Sigma_N$
   $\forall$ A : : = X producción de la gramática, en AP se hace: (q , X) $\in f$ (q , λ , A)
2. $\forall$ a $\in \Sigma_T$ entonces, (q , λ) $\in f$ (q , a , a)

SOLUCION MÉTODO 2:

Se va a construir un AP que acepte el mismo lenguaje generado por la gramática utilizada para el método 1:

G = { { a, b }, { S, A, B, C }, $\mathcal{P}$, S } $\mathcal{P} \equiv$

S : : = 0AC | 0BCC | 0C | 0CC | λ A : : = 0AC | 0C
B : : = 0BCC | 0CC C : : = 1

AP = { Σ_T , { $\Sigma_N \cup \Sigma_T$ } , { q } , S , q , f , $\varnothing$ } , AP = { { a, b }, { a, b, S, A, B } , { q } , S , q , f , $\varnothing$ }

Aplicamos el ALGORITMO para obtener los movimientos del AP.

f (q , λ , S) = { (q , 0AC) , (q , 0BCC) , (q , 0C) , (q , 0CC) , (q , λ) }
f (q , λ , A) = { (q , 0AC) , (q , 0C) }
f (q , λ , B) = { (q , 0BCC) , (q , 0CC) }
f (q , λ , C) = { (q , 1) }
f (q , 0 , 0) = { (q , λ) }
f (q , 1 ,1) = { (q , λ) }

RECONOCIMIENTO AP (0011): (q , 0011 , S) $\vdash$ (q , 0011 , 0AC) $\vdash$ (q , 011 , AC) $\vdash$ (q , 011 , 0CC) $\vdash$ (q , 11 , CC) $\vdash$ (q , 11 , 1C) $\vdash$ (q , 1 , C) $\vdash$ (q , 1 , 1) $\vdash$ (q , λ , λ)

RECONOCIMIENTO AP (001111): (q , 001111 , S) $\vdash$ (q , 001111 , 0BCC) $\vdash$ (q , 01111 , BCC) $\vdash$ (q , 01111 , 0CCCC) $\vdash$ (q , 1111 , CCCC) $\vdash$ (q , 1111 , 1CCC) $\vdash$ (q , 111 , CCC) $\vdash$ (q , 111 , 1CC) $\vdash$ (q , 11 , CC) $\vdash$ (q , 11 , 1C) $\vdash$ (q , 1 , C) $\vdash$ (q , 1 , 1) $\vdash$ (q , λ , λ)

GENERACION G (0011): S $\rightarrow$ 0AC $\rightarrow$ 00CC $\rightarrow$ 001C $\rightarrow$ 0011

GENERACION G (001111): S $\rightarrow$ 0BCC $\rightarrow$ 00CCCC $\rightarrow$ 001CCC $\rightarrow$ 0011CC $\rightarrow$ 00111C $\rightarrow$ 001111

Facultad de Informática de Madrid

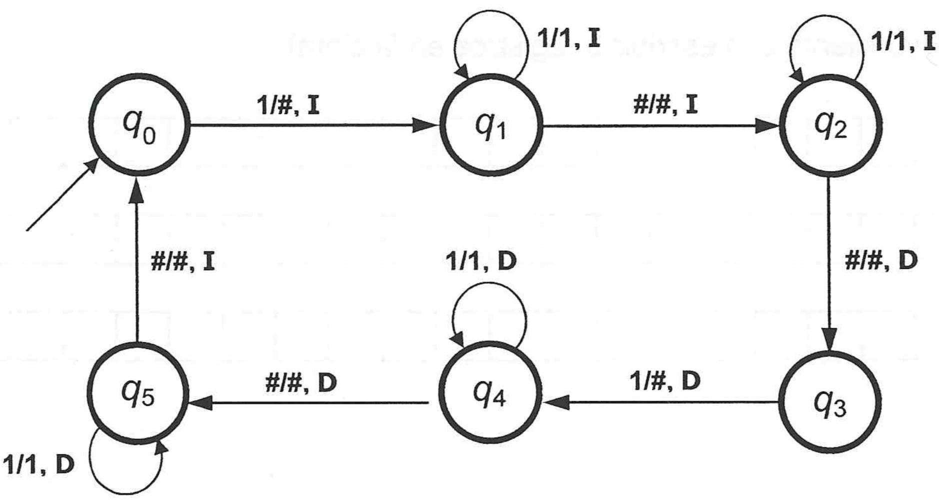
LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL – 2ª EVALUACIÓN (21 de enero de 2016)

Apellidos: SOLUCIÓN Nombre:

Ejercicio 2:

Sea la Máquina de Turing M definida según el siguiente grafo:



Y cuya configuración inicial es la siguiente: # x # y #, con la cabeza en q_0

Donde x e y son dos números enteros positivos codificados en unario cumpliéndose que $x \geq y$. M inicialmente está en el estado q_0 leyendo el último 1 de y.

a) ¿Qué función aritmética sobre los números x e y calcula M?. Mostrar el funcionamiento de M cuando el contenido inicial de su cinta y la posición de su cabeza de lectura-escritura son las siguientes: (2.5 puntos)

a.1) # 1 1 1 # 1 # (q_0)

a.2) # 1 1 # 1 1 # (q_0)

b) Escribir (y describir brevemente) el contenido inicial de la cinta de una Máquina de Turing Universal cuando simula a la máquina M y ésta recibe como entrada los números del apartado a.1). (2.5 puntos). Utilizar la siguiente codificación binaria:

$q_0 \equiv 000$; $q_1 \equiv 001$; $q_2 \equiv 010$, $q_3 \equiv 011$, $q_4 \equiv 100$, $q_5 \equiv 101$.

Desplazamiento a la izquierda. I ≡ 1; Desplazamiento a la derecha. D ≡ 0

c) Escribir (y describir brevemente) el contenido de la cinta de la Máquina de Turing Universal tras simular el primer movimiento que realiza la máquina M con la entrada del apartado a.1). (2.5 puntos).

d) Escribir (y describir brevemente) el contenido final de la cinta de la Máquina de Turing Universal cuando termine de simular a la máquina M con la entrada del apartado a.1). (2.5 puntos).

NOTA: Todos los apartados se responderán en la carilla de atrás.

30 minutos

Continuación ejercicio 2. RESPUESTAS. SOLUCIONES

Apartado a) M calcula $x-y$

a.1) $111\#q_0 1 \vdash 111 q_1 \#\# \vdash 11 q_2 1\# \vdash \cdots \vdash q_2 \# 111 \vdash q_3 111 \vdash \# q_4 11 \vdash \cdots \vdash 11 q_4 \# \vdash$
$\vdash 11\# q_5 \# \vdash 11 q_0 \#$ Para

a.2) $11\#1 q_0 1 \vdash 11\# q_1 1 \vdash 11 q_1 \# 1 \vdash \ldots \ldots \vdash \#\# q_0 \#\#\#$

Apartado b) (Es suficiente con escribir 3 registros en la cinta)

							q_0			1		q_0			1	q_1			#	I	
0 ...	1	1	1	0	*	∓	0	0	0	1	∓	0	0	0	1	0	0	1	0	1	∓ ...

0	0	1	1	0	0	1	1	1	∓	0	0	1	0	0	1	0	0	1	∓		...
q_1			1	q_1			1	I		q_1			#	q_2			#	I			

																					...

M está leyendo un 1 en q_0. Por eso, el REG inicial: $∓\underbrace{000}_{q_0}\underbrace{1}_{1}∓$
* en la celda que inicialmente lee M
Hay que reservar una celda en blanco (0) a la izqda para que la MTU puede simular a M.

Apartado c) (es suficiente con escribir sólo la parte de la cinta que cambia)

								q_1			#										
...	0	~~1~~	1	1	*	0	∓	0	0	1	0	∓	·	·							...

El primer movto que realiza M es $\delta(q_0,1)=(q_1,\#,I)$
Por tanto: el rep. inicial actualiza el estado. De q_0 a q_1. Se borra un 1 (donde estaba *
El * se recoloca en la celda de la izqda. El símbolo que hay en esa celda se guarda en la última celda del rep. inicial

Apartado d)

0 ...	0	1	1	*	0	∓	A	0	0	0	∓	A	A	A	B	A	A	B	A	B ∓ ...

		A's		B's		-	-	-	-											...

																				...

Cuando M se para se encuentra en la configuración: $11q_0\#$
La MTU se para porque no hay ningún registro que empiece por $\underbrace{000}_{q_0}0$
El módulo localizador marca y rechaza todos los registros con A's y B's
En la parte izqda hay en la cinta $\ldots\overline{11*0∓}\ldots$ que indica que M se para con el resultado de restar 111 y 1 Leyendo la celda en blanco de la dcha. (*)